HOME COPY

PCT/US01/46712

PCT

REQUEST

The undersigned requests that the present international application be processed according to the Patent Cooperation Treaty.

For receiving Office use only

PCT/US 01/46712
International Application No.

09 NOV 2001 (09.11.01)
International Filing Date

PCT INTERNATIONAL APPLICATION RO/US
Name of receiving Office and "PCT International Application"

Applicant's or agent's file reference *(if desired) (12 characters maximum)* 41016.P005

Box No. I TITLE OF INVENTION

Namespace Based Function Invocation

Box No. II APPLICANT ☐ This person is also inventor

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

BEA Systems, Incorporated
2315 North First Street
San Jose, California 95131
United States of America

Telephone No.

Facsimile No.

Teleprinter No.

Applicant's registration No. with the Office

State *(that is, country)* of nationality: ~~United States~~ ▲ US

State *(that is, country)* of residence: ~~United States~~ ▲ US

This person is applicant for the purposes of: ☐ all designated States ☒ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Box No. III FURTHER APPLICANT(S) AND/OR (FURTHER) INVENTOR(S)

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

BOSWORTH, Adam
934 SE 57th Street
Mercer Island, WA 98040
United States of America

This person is:
☐ applicant only
☒ applicant and inventor
☐ inventor only *(If this check-box is marked, do not fill in below.)*

Applicant's registration No. with the Office

State *(that is, country)* of nationality: ~~United States~~ ▲ US

State *(that is, country)* of residence: ~~United States~~ ▲ US

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☒ the United States of America only ☐ the States indicated in the Supplemental Box

☐ Further applicants and/or (further) inventors are indicated on a continuation sheet.

Box No. IV AGENT OR COMMON REPRESENTATIVE; OR ADDRESS FOR CORRESPONDENCE

The person identified below is hereby/has been appointed to act on behalf of the applicant(s) before the competent International Authorities as: ☒ agent ☐ common representative

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)*

AUYEUNG, Aloysius T.C.
COLUMBIA IP LAW GROUP, PC
10260 SW Greenburg Road, Suite 820
Portland, Oregon 97223
United States of America

Telephone No.
(503) 595-2800

Facsimile No.
(503) 595-2804

Teleprinter No.

Agent's registration No. with the Office
35,432

☐ **Address for correspondence:** Mark this check-box where no agent or common representative is/has been appointed and the space above is used instead to indicate a special address to which correspondence should be sent.

Continuation of Box No. III FURTHER APPLICANT(S) AND/OR (FURTHER) INVENTOR(S)

If none of the following sub-boxes is used, this sheet should not be included in the request.

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

BAU, David III
415 Howard Road
Gladwyne, Pennsylvania 19035
United States of America

This person is:
- ☐ applicant only
- ☒ applicant and inventor
- ☐ inventor only *(If this check-box is marked, do not fill in below.)*

Applicant's registration No. with the Office

State *(that is, country)* of nationality: ~~United States~~ ▲ US

State *(that is, country)* of residence: ~~United States~~ ▲ US

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☒ the United States of America only ☐ the States indicated in the Supplemental Box

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

VASILIK, Kenneth Eric
4911 163rd Ave., NE
Redmond, Washington 98052
United States of America

This person is:
- ☐ applicant only
- ☒ applicant and inventor
- ☐ inventor only *(If this check-box is marked, do not fill in below.)*

Applicant's registration No. with the Office

State *(that is, country)* of nationality: ~~United States~~ ▲ US

State *(that is, country)* of residence: ~~United States~~ ▲ US

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☒ the United States of America only ☐ the States indicated in the Supplemental Box

▲RO/US

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

This person is:
- ☐ applicant only
- ☐ applicant and inventor
- ☐ inventor only *(If this check-box is marked, do not fill in below.)*

Applicant's registration No. with the Office

State *(that is, country)* of nationality:

State *(that is, country)* of residence:

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

This person is:
- ☐ applicant only
- ☐ applicant and inventor
- ☐ inventor only *(If this check-box is marked, do not fill in below.)*

Applicant's registration No. with the Office

State *(that is, country)* of nationality:

State *(that is, country)* of residence:

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

☐ Further applicants and/or (further) inventors are indicated on another continuation sheet.

Box No. V DESIGNATION OF STATES *Mark the applicable check-boxes below; at least one must be marked.*

The following designations are hereby made under Rule 4.9(a):

Regional Patent

☒ **AP** **ARIPO Patent:** GH Ghana, GM Gambia, KE Kenya, LS Lesotho, MW Malawi, MZ Mozambique, SD Sudan, SL Sierra Leone, SZ Swaziland, TZ United Republic of Tanzania, UG Uganda, ZW Zimbabwe, and any other State which is a Contracting State of the Harare Protocol and of the PCT

☒ **EA** **Eurasian Patent:** AM Armenia, AZ Azerbaijan, BY Belarus, KG Kyrgyzstan, KZ Kazakhstan, MD Republic of Moldova, RU Russian Federation, TJ Tajikistan, TM Turkmenistan, and any other State which is a Contracting State of the Eurasian Patent Convention and of the PCT

☒ **EP** **European Patent:** AT Austria, BE Belgium, CH & LI Switzerland and Liechtenstein, CY Cyprus, DE Germany, DK Denmark, ES Spain, FI Finland, FR France, GB United Kingdom, GR Greece, IE Ireland, IT Italy, LU Luxembourg, MC Monaco, NL Netherlands, PT Portugal, SE Sweden, TR Turkey, and any other State which is a Contracting State of the European Patent Convention and of the PCT

☒ **OA** **OAPI Patent:** BF Burkina Faso, BJ Benin, CF Central African Republic, CG Congo, CI Côte d'Ivoire, CM Cameroon, GA Gabon, GN Guinea, GW Guinea-Bissau, ML Mali, MR Mauritania, NE Niger, SN Senegal, TD Chad, TG Togo, and any other State which is a member State of OAPI and a Contracting State of the PCT *(if other kind of protection or treatment desired, specify on dotted line)* .....

National Patent *(if other kind of protection or treatment desired, specify on dotted line)*:

☒ **AE** United Arab Emirates	☒ **GH** Ghana	☒ **MX** Mexico
☒ **AG** Antigua and Barbuda	☒ **GM** Gambia	☒ **MZ** Mozambique
☒ **AL** Albania	☒ **HR** Croatia	☒ **NO** Norway
☒ **AM** Armenia	☒ **HU** Hungary	☒ **NZ** New Zealand
☒ **AT** Austria	☒ **ID** Indonesia	☒ **PL** Poland
☒ **AU** Australia	☒ **IL** Israel	☒ **PT** Portugal
☒ **AZ** Azerbaijan	☒ **IN** India	☒ **RO** Romania
☒ **BA** Bosnia and Herzegovina	☒ **IS** Iceland	☒ **RU** Russian Federation
☒ **BB** Barbados	☒ **JP** Japan	
☒ **BG** Bulgaria	☒ **KE** Kenya	☒ **SD** Sudan
☒ **BR** Brazil	☒ **KG** Kyrgyzstan	☒ **SE** Sweden
☒ **BY** Belarus	☒ **KP** Democratic People's Republic of Korea	☒ **SG** Singapore
☒ **BZ** Belize		☒ **SI** Slovenia
☒ **CA** Canada	☒ **KR** Republic of Korea	☒ **SK** Slovakia
☒ **CH & LI** Switzerland and Liechtenstein	☒ **KZ** Kazakhstan	☒ **SL** Sierra Leone
☒ **CN** China	☒ **LC** Saint Lucia	☒ **TJ** Tajikistan
☒ **CO** Colombia	☒ **LK** Sri Lanka	☒ **TM** Turkmenistan
☒ **CR** Costa Rica	☒ **LR** Liberia	☒ **TR** Turkey
☒ **CU** Cuba	☒ **LS** Lesotho	☒ **TT** Trinidad and Tobago
☒ **CZ** Czech Republic	☒ **LT** Lithuania	
☒ **DE** Germany	☒ **LU** Luxembourg	☒ **TZ** United Republic of Tanzania
☒ **DK** Denmark	☒ **LV** Latvia	☒ **UA** Ukraine
☒ **DM** Dominica	☒ **MA** Morocco	☒ **UG** Uganda
☒ **DZ** Algeria	☒ **MD** Republic of Moldova	☒ **US** United States of America
☒ **EC** Ecuador		
☒ **EE** Estonia	☒ **MG** Madagascar	☒ **UZ** Uzbekistan
☒ **ES** Spain	☒ **MK** The former Yugoslav Republic of Macedonia	☒ **VN** Viet Nam
☒ **FI** Finland		☒ **YU** Yugoslavia
☒ **GB** United Kingdom	☒ **MN** Mongolia	☒ **ZA** South Africa
☒ **GD** Grenada	☒ **MW** Malawi	☒ **ZW** Zimbabwe
☒ **GE** Georgia		

Check-boxes below reserved for designating States which have become party to the PCT after issuance of this sheet:

☐ ..... ☐ ..... ☐ .....

☐ ..... ☐ ..... ☐ .....

Precautionary Designation Statement: In addition to the designations made above, the applicant also makes under Rule 4.9(b) all other designations which would be permitted under the PCT except any designation(s) indicated in the Supplemental Box as being excluded from the scope of this statement. The applicant declares that those additional designations are subject to confirmation and that any designation which is not confirmed before the expiration of 15 months from the priority date is to be regarded as withdrawn by the applicant at the expiration of that time limit. *(Confirmation (including fees) must reach the receiving Office within the 15-month time limit.)*

Sheet No. .....

Box No. VIII (iv) DECLARATION: INVENTORSHIP (only for the purposes of the designation of the United States of America)

The declaration must conform to the following standardized wording provided for in Section 214; see Notes to Boxes Nos. VIII, VIII (i) to (v) (in general) and the specific Notes to Box No. VIII (iv). If this Box is not used, this sheet should not be included in the request.

Declaration of Inventorship (Rules 4.17(iv) and 51*bis*.1(a)(iv)) for the purposes of the designation of the United States of America:

I hereby declare that I believe I am the original, first and sole (if only one inventor is listed below) or joint (if more than one inventor is listed below) inventor of the subject matter which is claimed and for which a patent is sought.

This declaration is directed to the international application of which it forms a part (if filing declaration with application).

This declaration is directed to international application No. PCT/..... (if furnishing declaration pursuant to Rule 26*ter*).

I hereby declare that my residence, mailing address, and citizenship are as stated next to my name.

I hereby state that I have reviewed and understand the contents of the above-identified international application, including the claims of said application. I have identified in the request of said application, in compliance with PCT Rule 4.10, any claim to foreign priority, and I have identified below, under the heading "Prior Applications," by application number, country or Member of the World Trade Organization, day, month and year of filing, any application for a patent or inventor's certificate filed in a country other than the United States of America, including any PCT international application designating at least one country other than the United States of America, having a filing date before that of the application on which foreign priority is claimed.

Prior Applications: 60/248,015 .....

.....

I hereby acknowledge the duty to disclose information that is known by me to be material to patentability as defined by 37 C.F.R. § 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the PCT international filing date of the continuation-in-part application.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name: BOSWORTH, Adam .....

Residence: Mercer Island, Washington .....
(city and either US state, if applicable, or country)

Mailing Address: 934 SE 57th Street .....
Mercer Island, Washington 98040 .....

Citizenship: United States .....

Inventor's Signature: ..... Date: MAR (illegible) .....
(if not contained in the request, or if declaration is corrected or added under Rule 26*ter* after the filing of the international application. The signature must be that of the inventor, not that of the agent)

(of signature which is not contained in the request, or of the declaration that is corrected or added under Rule 26*ter* after the filing of the international application)

Name: BAU, David III .....

Residence: Gladwyne, Pennsylvania .....
(city and either US state, if applicable, or country)

Mailing Address: 415 Howard Road .....
Gladwyne, Pennsylvania 19035 .....

Citizenship: United States .....

Inventor's Signature: ..... Date: 5/13/2002 .....
(if not contained in the request, or if declaration is corrected or added under Rule 26*ter* after the filing of the international application. The signature must be that of the inventor, not that of the agent)

(of signature which is not contained in the request, or of the declaration that is corrected or added under Rule 26*ter* after the filing of the international application)

☐ This declaration is continued on the following sheet, "Continuation of Box No. VIII (iv)".

Box No. VIII (v) DECLARATION: NON-PREJUDICIAL DISCLOSURES OR EXCEPTIONS TO LACK OF NOVELTY

The declaration must conform to the standardized wording provided for in Section 215; see Notes to Boxes Nos. VIII, VIII (i) to (v) (in general) and the specific Notes to Box No. VIII (v). If this Box is not used, this sheet should not be included in the request.

Declaration as to non-prejudicial disclosures or exceptions to lack of novelty (Rules 4.17(v) and 51*bis*.1(a)(v)):

Name: VASILIK, Kenneth Eric
Residence: Redmond, Washington
Mailing Address: 4911 163rd Ave., NE
Citizenship: United States

Inventor's Signature: K. Eric Vasilik Date: 3/18/2002

☐ This declaration is continued on the following sheet, "Continuation of Box No. VIII (v)".

Box No. VI PRIORITY CLAIM

The priority of the following earlier application(s) is hereby claimed:

Filing date of earlier application *(day/month/year)*	Number of earlier application	Where earlier application is:		
		national application: country	regional application:* regional Office	international application: receiving Office
item (1) 10 November 2000 (10.11.00)	60/246,915	US		
item (2)				
item (3)				
item (4)				
item (5)				

☐ Further priority claims are indicated in the Supplemental Box.

The receiving Office is requested to prepare and transmit to the International Bureau a certified copy of the earlier application(s) *(only if the earlier application was filed with the Office which for the purposes of this international application is the receiving Office)* identified above as:

☐ all items ☒ item (1) ☐ item (2) ☐ item (3) ☐ item (4) ☐ item (5) ☐ other, see Supplemental Box

** Where the earlier application is an ARIPO application, indicate at least one country party to the Paris Convention for the Protection of Industrial Property or one Member of the World Trade Organization for which that earlier application was filed (Rule 4.10(b)(ii)):* ....

Box No. VII INTERNATIONAL SEARCHING AUTHORITY

Choice of International Searching Authority (ISA) *(if two or more International Searching Authorities are competent to carry out the international search, indicate the Authority chosen; the two-letter code may be used)*:

ISA / US

Request to use results of earlier search; reference to that search *(if an earlier search has been carried out by or requested from the International Searching Authority)*:

Date *(day/month/year)* Number Country *(or regional Office)*

Box No. VIII DECLARATIONS

The following **declarations** are contained in Boxes Nos. VIII (i) to (v) *(mark the applicable check-boxes below and indicate in the right column the number of each type of declaration)*: Number of declarations

☐ Box No. VIII (i)	Declaration as to the identity of the inventor	:
☐ Box No. VIII (ii)	Declaration as to the applicant's entitlement, as at the international filing date, to apply for and be granted a patent	:
☐ Box No. VIII (iii)	Declaration as to the applicant's entitlement, as at the international filing date, to claim the priority of the earlier application	:
☐ Box No. VIII (iv)	Declaration of inventorship (only for the purposes of the designation of the United States of America)	:
☐ Box No. VIII (v)	Declaration as to non-prejudicial disclosures or exceptions to lack of novelty	:

Box No. IX CHECK LIST; LANGUAGE OF FILING

This international application **contains**:

(a) the following number of **sheets in paper form**:

request (including declaration sheets)	:	6
description (excluding sequence listing part)	:	11
claims	:	5
abstract	:	1
drawings	:	5
Sub-total number of sheets	:	28
sequence listing part of description *(actual number of sheets if filed in paper form, whether or not also filed in computer readable form; see (b) below)*	:	
Total number of sheets	:	28

(b) sequence listing part of description filed in **computer readable form**

(i) ☐ only (under Section 801(a)(i))

(ii) ☐ in addition to being filed in paper form (under Section 801(a)(ii))

Type and number of carriers (diskette, CD-ROM, CD-R or other) on which the sequence listing part is contained *(additional copies to be indicated under item 9(ii), in right column)*: .....

This international application is **accompanied** by the following item(s) *(mark the applicable check-boxes below and indicate in right column the number of each item)*:

		Number of items
1. ☒	fee calculation sheet :	
2. ☐	original separate power of attorney :	
3. ☐	original general power of attorney :	
4. ☒	copy of general power of attorney; reference number, if any: ..... :	
5. ☐	statement explaining lack of signature :	
6. ☐	priority document(s) identified in Box No. VI as item(s): ..... :	
7. ☐	translation of international application into *(language)*: ..... :	
8. ☐	separate indications concerning deposited microorganism or other biological material :	
9. ☐	sequence listing in computer readable form (indicate also type and number of carriers (diskette, CD-ROM, CD-R or other))	
	(i) ☐ copy submitted for the purposes of international search under Rule 13*ter* only (and not as part of the international application) :	
	(ii) ☐ *(only where check-box (b)(i) or (b)(ii) is marked in left column)* additional copies including, where applicable, the copy for the purposes of international search under Rule 13*ter* :	
	(iii) ☐ together with relevant statement as to the identity of the copy or copies with the sequence listing part mentioned in left column :	
10. ☒	other *(specify)*: Form PTO-1382 ..... :	

Figure of the drawings which should accompany the abstract: 1

Language of filing of the international application: English

Box No. X SIGNATURE OF APPLICANT, AGENT OR COMMON REPRESENTATIVE

Next to each signature, indicate the name of the person signing and the capacity in which the person signs (if such capacity is not obvious from reading the request).

Aloysius T.C. AuYeung

(09.11.01)

For receiving Office use only

1. Date of actual receipt of the purported international application:	JC18 Rec'd PCT/PTO 09 NOV 2001	2. Drawings: ☐ received: ☐ not received:
3. Corrected date of actual receipt due to later but timely received papers or drawings completing the purported international application:		
4. Date of timely receipt of the required corrections under PCT Article 11(2):		
5. International Searching Authority (if two or more are competent): ISA / US	6. ☐ Transmittal of search copy delayed until search fee is paid	

For International Bureau use only

Date of receipt of the record copy by the International Bureau:

This sheet is not part of and does not count as a sheet of the international application.

PCT

FEE CALCULATION SHEET

Annex to the Request

For receiving Office use only

International Application No. PCT/US 01/46712

Date stamp of the receiving Office 09 NOV 2001 (09.11.01)

Applicant's or agent's file reference 41016.P005

Applicant: BEA Systems, Incorporated

CALCULATION OF PRESCRIBED FEES

1. TRANSMITTAL FEE	240 T	240
2. SEARCH FEE	700 S	700

International search to be carried out by US

(If two or more International Searching Authorities are competent to carry out the international search, indicate the name of the Authority which is chosen to carry out the international search.)

3. INTERNATIONAL FEE

Basic Fee

Where item (b) of Box No. IX applies, enter **Sub-total number of sheets** / Where item (b) of Box No. IX does not apply, enter **Total number of sheets**: 28

b1 first 30 sheets	382 b1	382
b2 ____ (number of sheets in excess of 30) x ____ (fee per sheet) =	0 b2	—
b3 additional component (only if sequence listing part of description is filed in computer readable form under Section 801(a)(i), or both in that form and on paper, under Section 801(a)(ii)): 400 x ____ (fee per sheet) =	0 b3	
Add amounts entered at b1, b2 and b3 and enter total at B	382 B	382

Designation Fees

The international application contains 87 designations.

6 (number of designation fees payable *(maximum 6)*) x 82 (amount of designation fee) =	492 D	492
Add amounts entered at B and D and enter total at I	874 I	874

(Applicants from certain States are entitled to a reduction of 75% of the international fee. Where the applicant is (or all applicants are) so entitled, the total to be entered at I is 25% of the sum of the amounts entered at B and D.)

4. FEE FOR PRIORITY DOCUMENT *(if applicable)*	0 P	15
5. TOTAL FEES PAYABLE — Add amounts entered at T, S, I and P, and enter total in the TOTAL box	USD 1,814 TOTAL	1829

☐ The designation fees are not paid at this time.

MODE OF PAYMENT

☐ authorization to charge deposit account (see below)	☐ postal money order	☐ cash	☐ coupons
☒ cheque	☐ bank draft	☐ revenue stamps	☒ other *(specify)*: Postcard

AUTHORIZATION TO CHARGE (OR CREDIT) DEPOSIT ACCOUNT

(This mode of payment may not be available at all receiving Offices)

Receiving Office: RO/ US

Deposit Account No.: 501569

Date: 09 November 2001

Name: Aloysius T.C. AuYeung

Signature:

☐ Authorization to charge the total fees indicated above.

☒ *(This check-box may be marked only if the conditions for deposit accounts of the receiving Office so permit)* Authorization to charge any deficiency or credit any overpayment in the total fees indicated above.

☐ Authorization to charge the fee for priority document.

PCT
GENERAL POWER OF ATTORNEY
(for several international applications filed under the Patent Cooperation Treaty)
(PCT Rule 90.5)

The undersigned person(s); (Family name followed by given name, for a legal entity, full official designation. The address must include postal code and name of country.)

Robert F. Donohue
Senior Vice President, General Counsel and Secretary
BEA Systems, Inc.
2315 North First Street
San Jose, California 95131
United States of America

● ○

Hereby appoint(s) the following person as ● agent ○ common representative

Name and address
(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country).

AUYEUNG, Aloysius T.C.
DIEHL, Robert A.
KLINDTWORTH, Jason K.
WATT, Robert T.
CHANG, Robert H.
WERNER, Raymond J.

COLUMBIA IP LAW GROUP, PC
10260 SW Greenburg Road, Suite 820
Portland, Oregon 97223
United States of America

To represent the undersigned before

● all the competent International Authorities
○ the International Searching Authority only
○ the International Preliminary Examining Authority only

in connection with any and all international applications filed by the undersigned with the following Office (US) United States as receiving Office and to make or receive payments on behalf of the undersigned.

Signature(s) (where there are several persons, each of them must sign; next to each signature, indicate the name of the person signing and the capacity in which the person signs, if such capacity is not obvious from reading this power.

Robert F. Donohue,
Senior Vice President, General Counsel and Secretary

9 Nov 01
Date

Nam space Based Function Invocation

Related Applications

This non-provisional application is related to and claims priority to provisional application number 60/246,915 entitled "A Data Processing Method Employing Cell Based Data Flow Description", filed on November 10, 2000, which is hereby fully incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of data processing. More specifically, the present invention relates to the employment of namespaces to facilitate local and remote function invocation.

2. Background Information

Ever since the invention of the first computer, computer scientists have continuously tried to improve the productivity of programmers, such that more applications can be developed with fewer resources to take advantage of the continuous advancements being made in the art of computer and related technologies. First assembler languages were developed to replace machine languages. Then, high level languages, such as FORTRAN, COBOL, PL/I and so forth, were developed to further improve the productivity of programmers. Development of high level languages were followed by structured languages such as Pascal and C, and then object oriented programming languages such as C++. To facilitate development of the Internet and the World Wide Web, "new" languages such as the Hypertext Markup Language (HTML), Java, Javascript, Perl and CGI were developed.

Most languages also support function calls, which may be user-defined functions or pre-packaged functions, such as those included with the languages' runtime libraries. However, in the earlier days, to be invocable, the

functions must exist on the same system as the callers of the functions. Later on, to facilitate development of client-server computing, remote procedure call (RPC), a messaging protocol, was developed to facilitate a program executing on one computer to remotely invoke and access the service of a function on another computer. More recently, to facilitate development of web based applications, and employment of object oriented programming techniques, remote method invocation (RMI) was developed to facilitate interacting with objects on a server.

With the advance of public networks, such as the Internet, numerous function resources exist in different parts of the networks, written in different languages on different platforms. It is desirable to be able to invoke these function resources, regardless of their location, host language or host platform without limitation to the relationship between the invoking computing device and the function hosting computing device. Thus, an approach that can efficiently realize this potential is desired.

SUMMARY OF THE INVENTION

A data processing representation is read and parsed. A namespace declaration is recognized and identified as "executable." An expression is recognized. A name (declared within the executable namespace) within the expression having zero or more additional data representations is also recognized. In response, functions corresponding to the name and additional data representations are recursively resolved, and caused to be invoked and executed, with the execution results of the inner functions successively provided for use in the execution of the outer functions.

For each function, the function or a creator to create the function is located and instantiated based on the namespace and the name. In one embodiment, the declaration includes a URI to help locate functions. If a creator of the function is located and instantiated, the function is created using the function creator, and then the created function is instantiated. Upon instantiation of the referenced function, the function is executed.

In one embodiment, an execution engine is provided to effectuate the namespace basedfunction invocation. For the embodiment, the execution engine first looks for loadable Java class, thereafter a compilable Java resource, next, an XSLT style sheet and finally, a Java class factory. Further, for the embodiment, if the function is resolved to be a remote function, the execution engine creates a local proxy, causing the function to be remotely invoked and executed, passing the remote function with invocation parameters, where applicable.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

Figure 1 illustrates an overview of the namespace based function invocation the present invention, in accordance with one embodiment;

Figure 2a-2c illustrate the relevant operational flow of the execution engine of **Fig. 1**, in accordance with one embodiment; and

Figure 3 illustrates a computer system suitable for use to practice the present invention, in accordance with one embodiment.

DETAILED DESCRIPTION OF THE INVENTION

The present invention includes namespace based invocation of local or remote functions. In the following description, various aspects of the present invention will be described. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all aspects of the present invention. For purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a thorough understanding of the present invention. However, it will also be apparent to one skilled in the art that the present invention may be practiced without the specific details. In other

instances, well known features are omitted or simplified in order not to obscure the present invention.

Parts of the description will be presented in terms of operations performed by a computer system, using terms such as data, values, tags, references, and the like, consistent with the manner commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. As well understood by those skilled in the art, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, and otherwise manipulated through mechanical and electrical components of the computer system; and the term computer system includes general purpose as well as special purpose data processing machines, systems, and the like, that are standalone, adjunct or embedded.

Various operations will be described as multiple discrete steps in turn, in a manner that is most helpful in understanding the present invention, however, the order of description should not be construed as to imply that these operations are necessarily order dependent. In particular, these operations need not be performed in the order of presentation.

Overview

Referring now to **Figure 1**, wherein a block diagram illustrating an overview of the namespace based function invocation of the present invention, in accordance with one embodiment. As illustrated, in accordance with the present invention, a computing environment **102** is provided with an execution engine **104** equipped to recognize, construct, and effectuate invocation references in a data representation **106** to one or more local or remote functions **114**. As will be described in more details below, locations **112** of functions **114** are resolved by their affiliations to qualified names **110** declared in namespaces **108**, which may be local or remote. As a result, function invocation is advantageously streamlined.

In general, except for the teachings of the present invention incorporated in execution engine **104**, and the exploitation of these abilities to process data representations **106**, data representations **106** are intended to

represent a broad range of data representation methodologies known in the art, and execution engine **104** is intended to represent a broad range of the corresponding engines in support of these methodologies. Typically, each execution engine **104** is designed to support a particular manner of representing data or a particular programming language, although in different embodiments, execution engine **104** may support multiple data representation methodologies/programming languages. Similarly, except for the ability of execution engine **104** being able to invoke functions **114** affiliated with qualified names **110** declared in namespaces **108,** functions **114** and their locations **112** are also intended to represent a broad range of these elements known in the art.

Further, computing environment **102** may be disposed in a single or multi-processor system, or a collection of networked systems. In the case of networked systems, the systems may be networked locally, or across a number of private and/or public networks, including the Internet.

The manner in which data representations **106** reference functions **114** affiliated with qualified names **110** declared in namespaces **108**, and the manner in which engine **104** recognizes, constructs and effectuates these invocations will be described in turn below.

Invocation of Functions Affiliated with Namespaces

In one embodiment, the present invention contemplates that a data representation **106** would associate a qualified name **110** with a function **114** by first declaring a namespace as "executable", using a specially formed URI, with e.g. the following syntax

<math:add xmlns:math="x://bestuniversity.edu/mathdept/mathlibr/">

where "xmlns" specifies an XML namespace is being declared

"math" is the prefix of the XML namespace, used to qualify names in this namespace.

"x://bestuniversity.edu/mathdept/mathlibr/" is a the unique name of the namespace.

"x:" is a special URI scheme recognized by the corresponding execution engine as identifying the XML namespace (associated with the prefix "math") being declared as executable, and

"bestuniversity.edu" is the URI authority or hostname and "/mathdept/mathlibr/" is a relative path. These are used to locate the functions associated with the qualified names in the XML namespace being declared.

"math:add" is a qualified name indicating "add" is the name of a function within the executable namespace "x://bestuniversity.edu/mathdept/mathlibr/".

The present invention further contemplates that the execution engine **104** would then process each data representation **106** associated with (e.g., nested within) the qualified name **110** recursively, concatenate the results together in order of occurrence, pass the concatenated results to a function **114** affiliated with the qualified name **110** and replace the qualified name with the result returned by the function, with e.g. the following syntax

```
<math:squareroot xmlns:math="x://bestuniversity.edu/mathdept/mathlibr/">
    <math:add>
        <math:square>4</math:square>
        <math:square>3</math:square>
    </math:add>
</math:squareroot>
```

By virtue of the namespace prefix "math", the names "squareroot" "add" and "square" are interpreted as references to a "squareroot", an "add" and a "square" function within namespace "x://bestuniversity.edu/mathdept/mathlibr/". Together, the statements operate to "square" the values 3 and 4, add the results together, and then generate the square root of the sum (i.e., 5), using the "squareroot", "add" and "square" functions associated with the qualified names in the declared namespace. Thus, math functions (or other functions of the like) available on the network from various third parties, may be easily invocable following the described approach.

Execution Engine

Figures 2a-2c illustrate the operational flow of the relevant aspects of execution engine **104** in accordance with one embodiment. More specifically, **Fig. 2a** illustrates the general operational flow of execution engine **104** for handling an invocation of a function, and **Fig. 2b-2c** illustrate the operation flow of execution engine **104** in resolving each of the nested functions (if any) within the function, and the function itself, being invoked. Recall from earlier description, the function, including each of the nested functions (if any), is affiliated with a qualified name **110** declared in a namespace **108**. Further, the embodiment, assumes, execution engine **104**, like other conventional execution engines of prior art data representations, upon invocation, would parse and interpret the statements of data representation **106**. Thus, as illustrated in **Fig. 2a**, for the embodiment, upon encountering an expression referencing a function with zero or more nested functions, execution engine **104** first locates a first "branch" with nested function(s), starting with the "topmost branch", block **201**. (Note that usage of the term "branch" and "topmost" are merely referential with respect to a "horizontal" view of the nesting structure, i.e. A → B → C, for ease of understanding). Next, execution engine **104** locates the innermost function along the selected branch within the function being invoked, block **202**. Naturally, if the there is no nested function within the function being invoked, the function itself is considered the innermost function located on the topmost branch. Upon locating the innermost function, execution engine **104** resolves the location of the function, and causes the function to be invoked and executed, block **203**.

Then, execution engine **104** determines if there is another outer function, block **204**. If so, execution engine **104** further determines if the function has other nested functions along other branches, block **205**. If not, the process continues back at block **203**, with execution engine **104** resolving and invoking the next outer function, passing the execution result of the previous invoked function(s) to the next outer function being invoked.

Back at block **205**, if the function is determined to have additional

nested functions along other "lower" branches, execution engine **104** recursively perform process **200** until all nested functions along these "lower" branches have been recursively invoked and executed, block **206**.

The process continues as earlier described until the outermost function has also been resolved and invoked. At such time, the recursively generated execution result becomes the final execution result.

Figures 2b-2c illustrate the manner in which execution engine **104** resolves a function affiliated with a qualified name prefix:name. As illustrated, execution engine **104** would first search for an invocable Java Class based on the declared namespace, and attempt to load and instantiate the referenced Java Class, block **204**. In the above example, execution engine **104** would use the qualified name and URI included in the namespace declaration, and construct a fully qualified name for the assumed Java Class, e.g. in the case of the "squareroot" function, "edu.bestuniversity.mathedept,mathlibr.squareroot", and attempt to load the Java Class from a classpath derived from the URI.

At block **205**, execution engine **104** would determine if the attempt loading was successful or e.g. an error code was returned. If the loading attempt was successful, execution engine **104** would instantiate the so loaded Java Class, block **206**, and execute the function accordingly. If not, for the embodiment, execution engine **104** would next infer the reference as a reference to a compilable Java resource based on the declared namespace, and attempt to retrieve the referenced Java resource for compilation, block **208**. In the above example, execution engine **104** would use the qualified name and URI included in the namespace declaration, and construct a path name for the assumed Java resource, e.g. in the case of the "squareroot" function, "/mathedept/mathlibr/squareroot.java", and attempt to retrieve the compilable Java resource from a search path derived from the URI.

As before, at block **210**, execution engine **104** would determine if the attempt retrieval was successful or e.g. an error code was returned. If the retrieval attempt was successful, execution engine **104** would compile the retrieved Java resource, block **212**, and instantiate the compiled code, block

206. Thereafter, the function is executed accordingly. If not, for the embodiment, execution engine **104** would next infer the reference as a reference to an Extensible Stylesheet Language Transformation (XSLT) based on the declared namespace, and attempt to retrieve the referenced XSLT sheet, block **214**. In the above example, execution engine **104** would use the qualified name and URI included in the namespace declaration, and construct a path name for the assumed XSLT sheet, e.g. in the case of the "squareroot" function, "/mathedept/mathlibr/squareroot.xslt", and attempt to retrieve the XSLT sheet from a search path derived from the URI.

At block **216**, execution engine **104** would determine if the attempt retrieval was successful or e.g. an error code was returned. If the retrieval attempt was successful, execution engine **104** would invoke the retrieved XSLT sheet, block **218**, and execute the function accordingly. If not, for the embodiment, execution engine **104** would next infer the reference as a reference to a Java Factory, and attempt to retrieve the referenced Java Factory, block **220**. In the above example, execution engine **104** would use the qualified name and URI included in the namespace declaration, and construct a fully qualified name for the assumed Java class factory, e.g. in the case of the "squareroot" function, "edu.bestuniversity.mathedept.mathlibrFactory", and attempt to retrieve the Java class factory from a class path derived from the URI.

Again, at block **222**, execution engine **104** would determine if the retrieval attempt was successful or e.g. an error code was returned. If the retrieval attempt was successful, execution engine **104** would instantiate the Java class factory, block **224**, and generate the Java Class using the instantiated Java class factory accordingly, block **226**. Thereafter, the referenced function would be executed accordingly.

If not, for the embodiment, execution engine **104** would next infer the reference as a reference to a remote function. Accordingly, execution engine **104** creates a local proxy, and attempts to call the function remotely, including passing any applicable invocation parameters to the remote function, if

applicable, block **230**. In the above example, execution engine **104** would use the URI included in the namespace declaration, to create a local proxy function for invoking the remote function, e.g. in the case of the "squareroot" function, a proxy function would be created to send the entire <math:squareroot> element to "bestuniversity.edu/mathedept/mathlibr" for remote evaluation (e.g., using HTTP POST), and return the remote response as the proxy function result. Thereafter, the referenced proxy function would be executed accordingly.

While for ease of understanding, the above description has enumerated only Java class, XSLT sheets, Java factory, and so forth, the present invention is not so limited. The present invention may be practiced with more or less programming methodologies/languages, including but not limited to JavaScript, XML files, Xsheets, and so forth. Xsheet is a cell based data processing methodology, which is the subject matter of U.S. patent application number <to be insert>, entitled "Cell Based Data Processing", filed on <insert date>, which is a non-provisional application of the earlier enumerated U.S. provisional patent application 60/246,915. Readers are referred to the 'xxx application for further details.

Accordinglyfunctions written in a variety of languages, on a variety of host platforms, located at different locations of interconnected networks may be remotely invoked, based on their affiliation to namespaces, making it a lot easier for their access and invocation, as compared to prior art approaches.

Example Computer System

Figure 3 illustrates a computer system suitable for use to practice the present invention, in accordance with one embodiment. As shown, computer system **300** includes one or more processors **302** and system memory **304**. Additionally, computer system **300** includes mass storage devices **306** (such as diskette, hard drive, CDROM and so forth), input/output devices **308** (such as keyboard, cursor control and so forth) and communication interfaces **310** (such as network interface cards, modems and so forth). The elements are coupled to each other via system bus **312**, which represents one or more

buses. In the case of multiple buses, they are bridged by one or more bus bridges (not shown). Each of these elements performs its conventional functions known in the art. In particular, system memory **304** and mass storage **306** are employed to store a working copy and a permanent copy of the programming instructions implementing the execution engine with the namespace based function invocation support. The permanent copy of the programming instructions may be loaded into mass storage **306** in the factory, or in the field, as described earlier, through a distribution medium (not shown) or through communication interface **310** (from a distribution server (not shown). The constitution of these elements **302-312** are known, and accordingly will not be further described.

Conclusion and Epilogue

Thus, it can be seen from the above descriptions, a novel method and apparatus for effectuating function invocation has been described. While the present invention has been described in terms of the above illustrated embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. The present invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of restrictive on the present invention.

CLAIMS

What is claimed is:

1. A method of computing comprising:

reading and parsing a data processing representation;

recognizing a declaration reference to an executable namespace;

recognizing an expression referencing a function of the executable namespace;

instantiating the referenced function or a function creator to create the function, then instantiate the created function; and

evaluating the expression using the instantiated function.

2. The method of claim 1, wherein said declaration includes a path in said executable namespace to be followed to locate functions of the executable namespace; and said instantiation comprises following said path to locate said referenced function or the function creator of the referenced function.

3. The method of claim 2, wherein said instantiating comprises

determining if a loadable Java class exists under a fully qualified name formed with said path and said referenced function; and

if the loadable Java class exists under the fully qualified name, instantiating said loadable Java class following said path.

4. The method of claim 2, wherein said instantiating comprises

determining if a loadable resource exists under a class path formed with said path said referenced function, and a class name; and

if the loadable resource exists under the class path, retrieving said loadable resource following said path, compiling said retrieved resource, and instantiating said compiled resource.

5. The method of claim 2, wherein said instantiating comprises

determining if a loadable XSLT style sheet exists under a class path formed with said path said referenced function, and an XSLT style sheet extension; and

if the loadable resource exists under the class path, retrieving said loadable XSLT style sheet following said class path, and calling said XSLT style sheet as a function section.

6. The method of claim 2, wherein said instantiating comprises

determining if a loadable resource exists under a class path formed with said path and a function creator name of said function; and

if the loadable resource exists under the class path, retrieving said loadable resource following said path, creating said function using said loadable resource, and instantiating said created function.

7. The method of claim 1, wherein said instantiating comprises first determining if a loadable Java class corresponding to the referenced function exists, and if not, whether a compilable resource corresponding to the referenced function exists.

8. The method of claim 1, wherein said instantiating comprises first determining if a Java resource corresponding to the referenced function in executable or compilable exists, and if not whether an XSLT style sheet resource corresponding to the referenced function exists.

9. The method of claim 1, wherein said instantiating comprises first determining if an XSLT style sheet corresponding to the referenced function resource exists, and if not whether a Java class factory corresponding to the referenced function exists.

10. The method of claim 1, wherein said method further comprises recognizing at least one other function nested within said referenced function of the expression, and said evaluation comprises recursively invoking and instantiating the nested functions.

11. An apparatus comprising:

at least one storage unit having stored thereon programming instructions designed to

read and parse a data processing representation;

recognize a declaration reference to an executable namespace;

recognize an expression referencing a function of the excutable namespace;

instantiate the referenced function or a function creator to create the function, then instantiate the created function; and

evaluate the expression using the instantiated function; and

at least one processor coupled to said at least one storage unit to execute said programming instructions.

12. The apparatus of claim 11, wherein said programming instructions are designed to recognize said declaration having including a path in said executable namespace to be followed to locate functions of the executable namespace; and to effectuate said instantiation by following said path to locate said referenced function or the function creator of the referenced function.

13. The apparatus of claim 12, wherein said programming instructions are designed to

determine if a loadable Java class exists under a fully qualified name formed with said path and said referenced function, and

if the loadable Java class exists under the fully qualified name, instantiate said loadable Java class following said path.

14. The apparatus of claim 12, wherein said programming instructions are designed to

determine if a loadable resource exists under a class path formed with said path said referenced function, and a class name, and

if the loadable resource exists under the class path, retrieve said loadable resource following said path, compile said retrieved resource, and instantiate said compiled resource.

15. The apparatus of claim 12, wherein said programming instructions are designed to

determine if a loadable XSLT style sheet exists under a class path formed with said path said referenced function, and an XSLT style sheet extension, and

if the loadable resource exists under the class path, retrieve said loadable XSLT style sheet following said class path, and call said XSLT style sheet as a function section.

16. The apparatus of claim 12, wherein said programming instructions are designed to

determine if a loadable resource exists under a class path formed with said path and a function creator name of said function, and

if the loadable resource exists under the class path, retrievesaid loadable resource following said path, create said function using said loadable resource, and instantiate said created function.

17. The apparatus of claim 11, wherein said programming instructions are designed to effectuate said instantiation by first determining if a loadable Java class corresponding to the referenced function exists, and if not, whether a compilable resource corresponding to the referenced function exists.

18. The apparatus of claim 11, wherein said programming instructions are designed to effectuate said instantiation by first determining if a Java resource

corresponding to the referenced function in executable or compilable exists, and if not whether an XSLT style sheet resource corresponding to the referenced function exists.

19. The apparatus of claim 11, wherein said programming instructions are designed to effectuate said instantiation by first determining if an XSLT style sheet resource corresponding to the referenced function exists, and if not whether a Java class factory corresponding to the referenced function exists.

20. The apparatus of claim 11, wherein said programming instructions are further designed to recognize one or more functions nested within said referenced function of the expression, and recursively invoke and instantiate the nested functions.

21. An apparatus comprising:

means for reading and parsing a data processing representation;

means for recognizing a declaration reference to an executable namespace, including a path within the exuctable namespace;

means for recognizing an expression referencing a function of the excutable namespace;

means for instantiating, following said path, the referenced function or a function creator to create the function, then instantiate the created function; and

means for evaluating the expression using the instantiated function.

Namespac Bas d Function Invocation

ABSTRACT OF THE DISCLOSURE

A data representation is read and parsed. A declaration referencing an executable namespace is recognized. An expression is recognized. A name (declared within the executable namespace) within the expression having zero or more additional data representations is also recognized. In response, functions corresponding to the name and additional data representations are recursively resolved, and caused to be invoked and executed, with the execution results of the inner functions successively provided for use in the execution of the outer functions. For each function, the function or a creator to create the function is instantiated. The instantiating is effectuated by following a path enumerated in the declaration for locating functions of the namespace. If a creator of the function is located and instantiated, the function is created using the function creator, and then the created function is instantiated. Upon instantiation, the referenced function is invoked and executed.

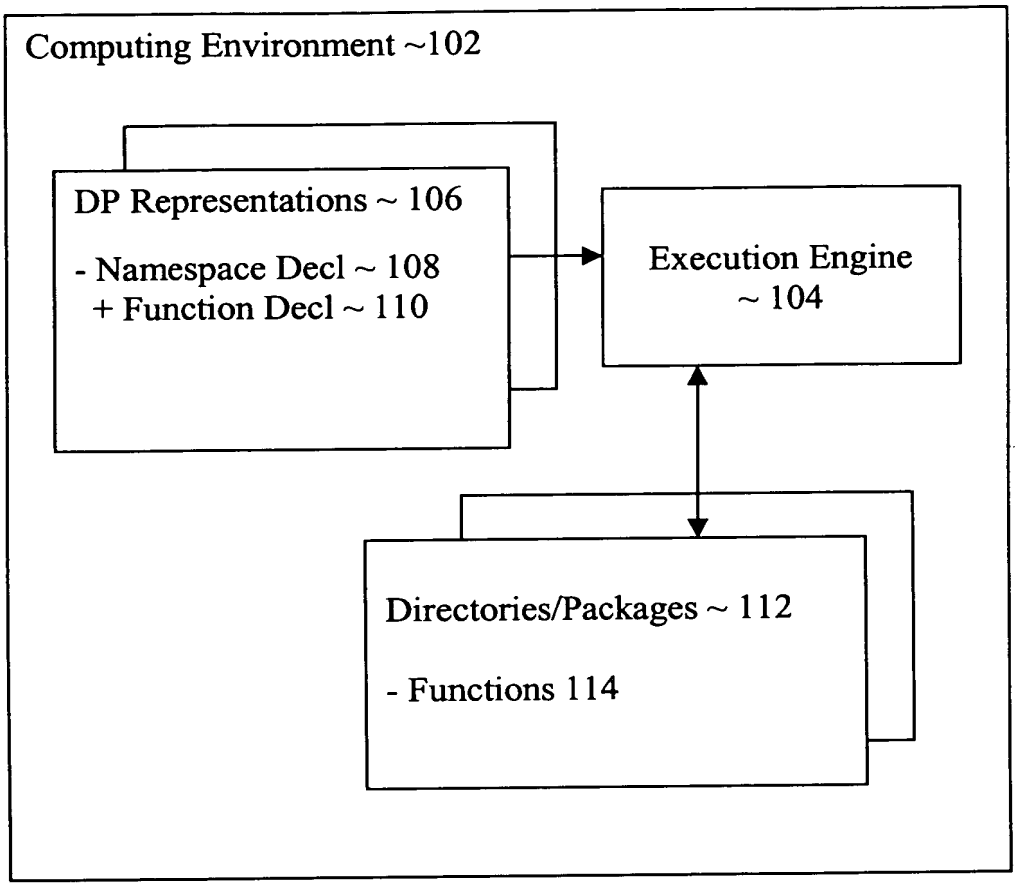
Computing Environment ~102
DP Representations ~ 106
- Namespace Decl ~ 108
+ Function Decl ~ 110
Execution Engine
~ 104
Directories/Packages ~ 112
- Functions 114

Figure 1

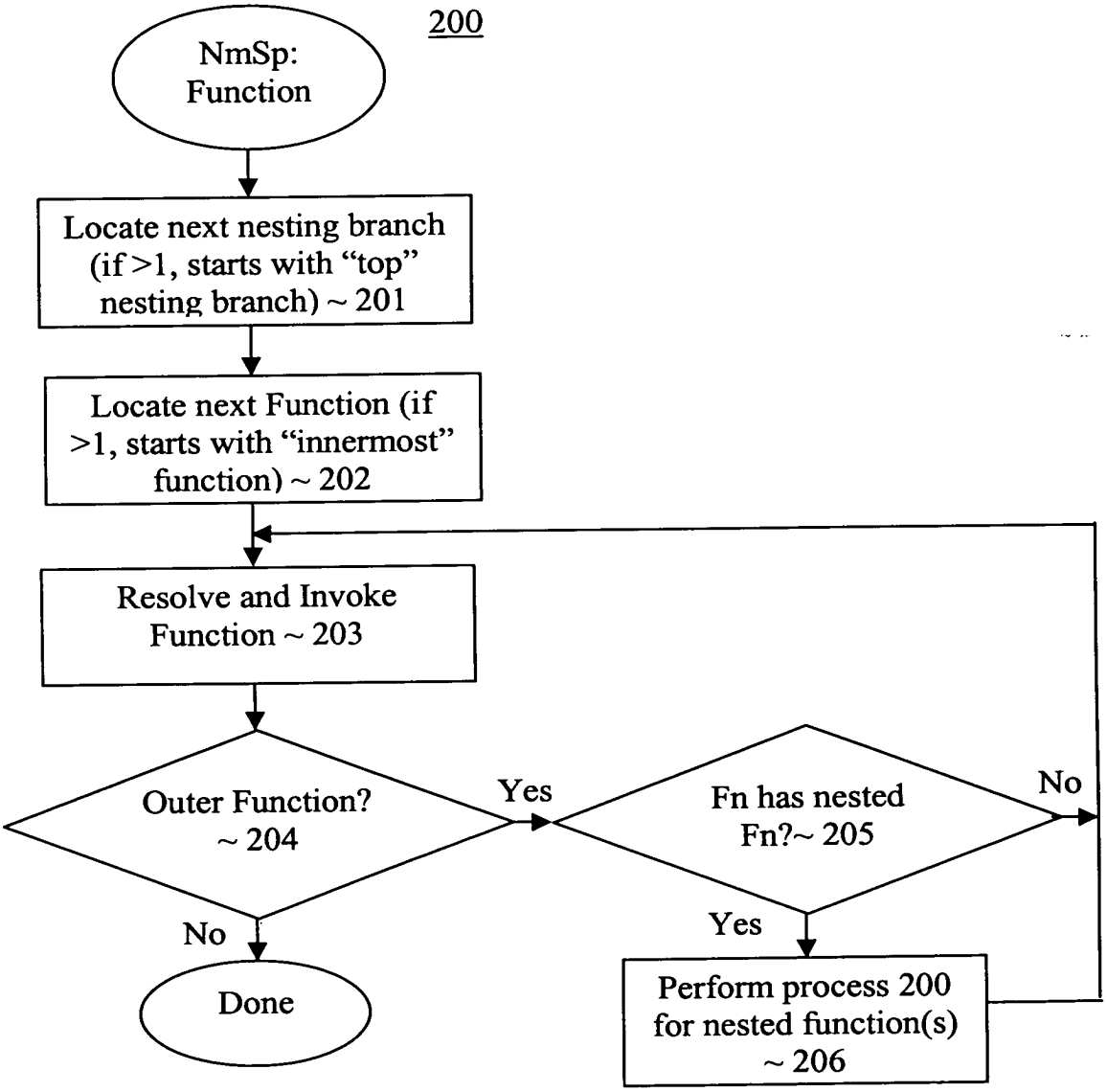
200
NmSp: Function
Locate next nesting branch (if >1, starts with "top" nesting branch) ~ 201
Locate next Function (if >1, starts with "innermost" function) ~ 202
Resolve and Invoke Function ~ 203
Outer Function? ~ 204
Yes
Fn has nested Fn?~ 205
No
No
Done
Yes
Perform process 200 for nested function(s) ~ 206

Figure 2a

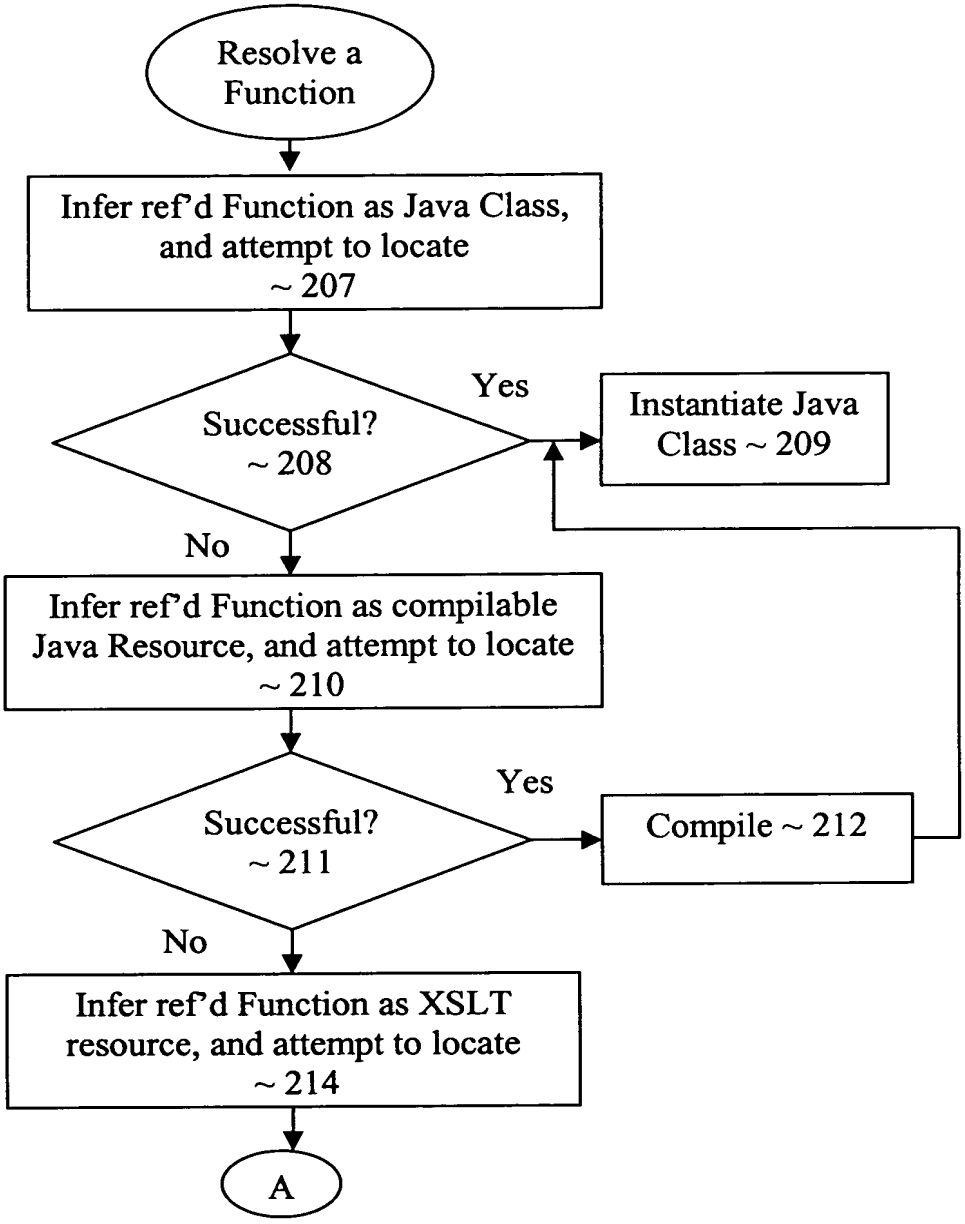
Resolve a Function
Infer ref'd Function as Java Class, and attempt to locate ~ 207
Successful? ~ 208
Yes
Instantiate Java Class ~ 209
No
Infer ref'd Function as compilable Java Resource, and attempt to locate ~ 210
Successful? ~ 211
Yes
Compile ~ 212
No
Infer ref'd Function as XSLT resource, and attempt to locate ~ 214
A

Figure 2b

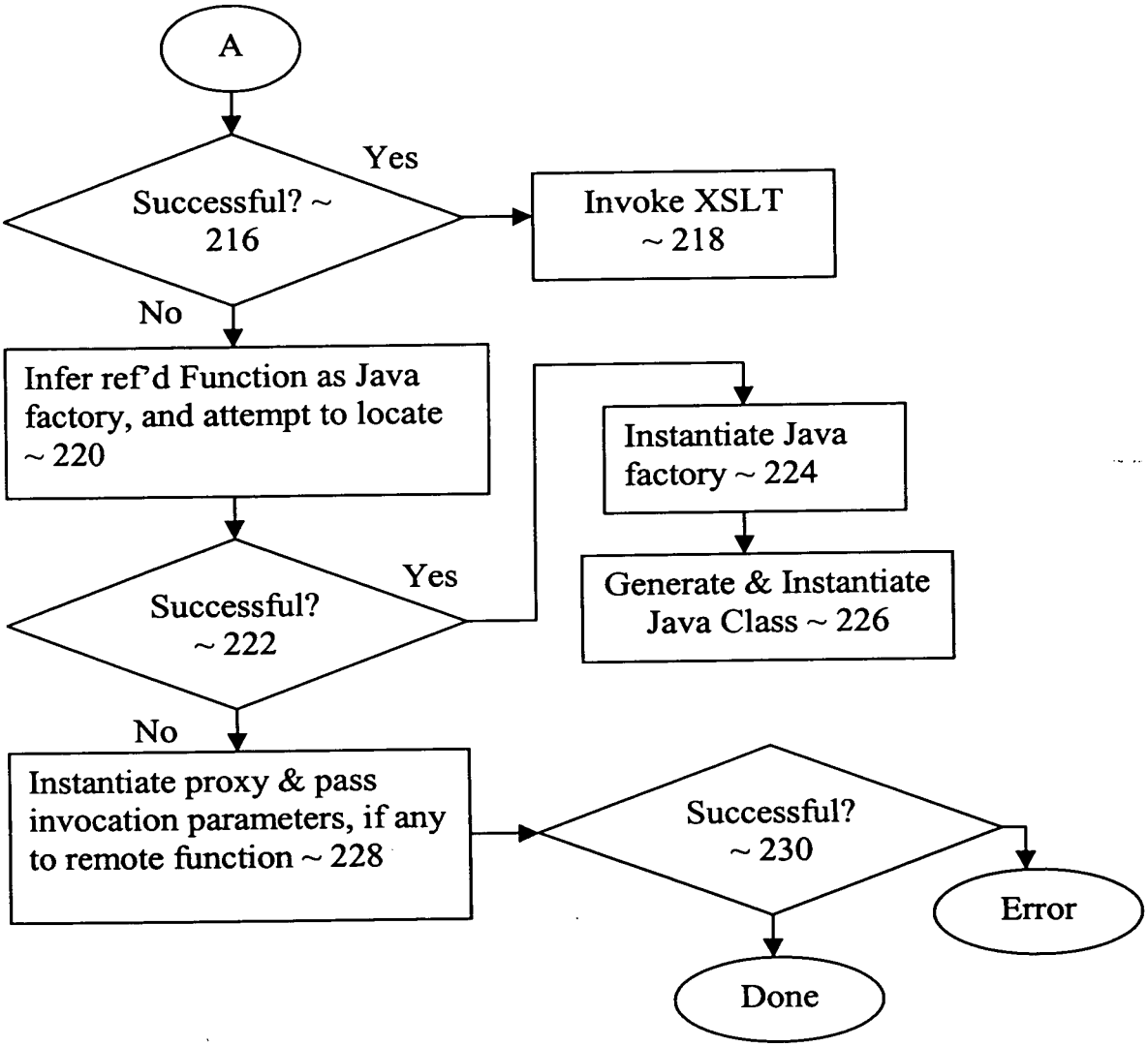
A
Successful? ~ 216
Yes
Invoke XSLT ~ 218
No
Infer ref'd Function as Java factory, and attempt to locate ~ 220
Instantiate Java factory ~ 224
Successful? ~ 222
Yes
Generate & Instantiate Java Class ~ 226
No
Instantiate proxy & pass invocation parameters, if any to remote function ~ 228
Successful? ~ 230
Error
Done

Figure 2c

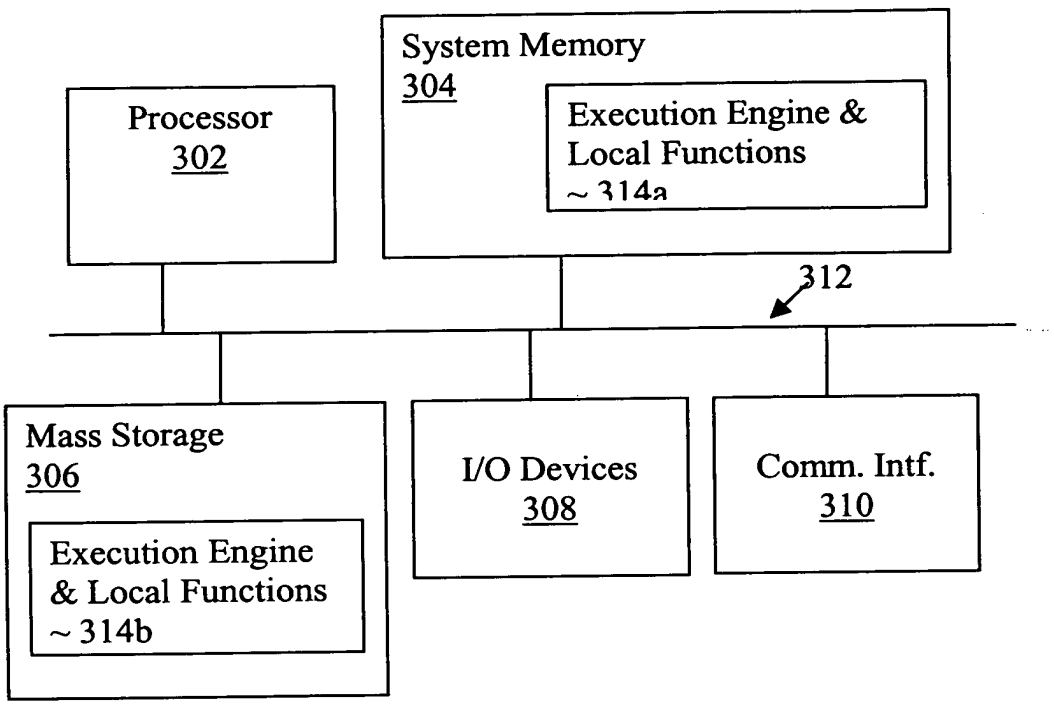
300
System Memory
304
Execution Engine &
Local Functions
~ 314a
Processor
302
312
Mass Storage
306
Execution Engine
& Local Functions
~ 314b
I/O Devices
308
Comm. Intf.
310

Figure 3